

Beheers document DevOps

DOCUMENTATIE VOOR DEVOPS PROJECT VAN GROEP 2
RICK BENIERS (STUDENT)

Inhoudsopgave

Inleiding.....	2
CI/CD Pipeline	3

Inleiding

Dit document is het beheer document voor de Igame applicatie van groep2 voor adsd te Windesheim.

In dit document zal de werk wijze van de pipeline worde uitgelegd en waar mogelijk reproduceerbaar worden opgeschreven.

Voor dit project is een GitHub repository gebruikt.

https://github.com/RickBeniers/I-Game_ADSD-Labs1/tree/master

Voor dit project is een dockerHub repository gebruikt.

<https://hub.docker.com/repository/docker/rickbeniers/adsd-2024-igame/general>

Voor dit project is gebruik gemaakt van codeclimate.com voor code analysis.

https://codeclimate.com/github/RickBeniers/I-Game_ADSD-Labs1

CI/CD Pipeline

Voor het opzetten en uitvoeren van een Continuous integration en Continuous development pipeline is gebruikt gemaakt van Github actions.

Github actions is een methode om via github automatische checks uit te laten voeren op git pushes en/of branch merges. Momenteel zijn er op github ongeveer 11.000 github actions beschikbaar voor publieke repositories.

Docker image build

Op de branches "Master" en "Development" op github wordt doormiddel van een github action een docker image gebouwd. Deze image wordt opgeslagen en automatisch verstuurd op de dockerhub repository. Na het succesvol uitvoeren van de image build workflow wordt dit gereflecteerd in een github badge in de Readme file op github.

App deployment

De app wordt gebouwd door een build artifact te maken en op te slaan op github. Dit is gekozen om hosting via een ander programma dan kubernetes te faciliteren in de toekomst.

Tijdens de deployment workflow worden ook twee commandos uitgevoerd die de image op dockerhub met de laatste versie uitvoeren op een kubernetes cluster doormiddel van de CLI cluster beheer programma kubectl. Na het succesvol uitvoeren van de deployment wordt dit gereflecteerd in een github badge in de Readme file op github.

Laravel code testing

Via een github action worden alle unittesten op de applicatie uitgevoerd doormiddel van het aanroepen van PHPUnit. Wanneer alle unit testen succesvol zijn uitgevoerd wordt dit gereflecteerd doormiddel van een github badge op de Readme file op github. Momenteel zijn er alleen standaard Laravel unittesten in het project opgenomen. Hiervoor is gekozen om tijd te besparen. Het schrijven van werkende Unittesten en infrastructuur testen waren een lage prioriteit vergeleken met het opbouwen van een werkende pipeline.

Code analysis

In het project wordt code analyse uitgevoerd door een github action die eveneens php unittesten laat uitvoeren. Hier wordt een rapport van gemaakt en dit rapport wordt verstuurd naar codeclimate.com dat gekoppeld is aan de github repository.